

N75-22157  
Unclas 18657  
63/76  
(NASA-CR-142673) PROCESSOR ARCHITECTURES  
UTILIZING MAGNETIC BUBBLE AND SEMI-CONDUCTOR  
MEMORIES Interim Report (Virginia Univ.)  
CSC 20L  
63 P HC \$4.25

PROCESSOR ARCHITECTURES UTILIZING  
MAGNETIC BUBBLE AND SEMI-CONDUCTOR MEMORIES

Interim Report

NASA Contract Number NSG-1067

Submitted to:

Mr. John Wells, Jr.  
Mail Stop 470  
National Aeronautics and Space Administration  
Langley Research Center  
Hampton, Virginia 23365

Submitted by:

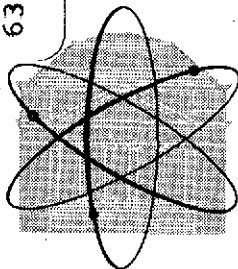
E. A. Parrish, Jr.  
Associate Professor

and

J. H. Aylor  
Research Engineer

SCHOOL OF ENGINEERING AND  
APPLIED SCIENCE

RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES



UNIVERSITY OF VIRGINIA  
CHARLOTTESVILLE, VIRGINIA 22901

Report No. EE-4027-102-75

April 1975

PRICES SUBJECT TO CHANGE

REPRODUCED BY  
NATIONAL TECHNICAL  
INFORMATION SERVICE  
U. S. DEPARTMENT OF COMMERCE  
SPRINGFIELD, VA. 22161

WAP

## **RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES**

The School of Engineering and Applied Science of the University of Virginia has long believed that strong research capabilities go hand in hand with effective teaching. Early in the development of its graduate training program, the School also recognized that men and women engaged in research should be as free as possible of the administrative chores involved in sponsored research. In 1959, therefore, the Research Laboratories for the Engineering Sciences (RLES) was established and assigned the administrative responsibility for such research within the School.

Currently, approximately 60 members of the faculty, who also teach at the undergraduate and graduate levels, and 30 additional professional engineers and scientists, whose primary responsibility is research, generate and conduct the investigations that make up a vigorous and wide-ranging program. The Director of RLES, a faculty member and active researcher himself, maintains familiarity with the support requirements of all research under way. He is aided by an RLES Academic Advisory Committee made up of one faculty representative from each academic department of the School. This Committee serves to inform RLES of the needs and perspectives of the research community.

In addition to administrative support, RLES is charged with providing technical assistance where it is needed. Because it is not practical for each department of the School to become self-sufficient in all phases of the supporting technology essential to present-day research, RLES makes services available through the following support groups: Machine Shop, Instrumentation, Facilities Services, Publications (including photographic facilities), and Computer Terminal Maintenance.

The purpose of RLES, then, is to provide administrative and technical assistance for sponsored research carried out within the School of Engineering and Applied Science of the University of Virginia. Such research has played an important part in the University's contribution to scientific knowledge and service to the community and continues the successful partnership of University, government, and industry.

For information on current programs and capabilities, write to Director, Research Laboratories for the Engineering Sciences, Thornton Hall, University of Virginia, Charlottesville, Virginia 22901.

Interim Report

The Design and Operation Manual  
of an Experimental Control Panel for a Low Cost  
Omega Navigational Receiver

## FOREWORD

This report presents the detailed design equations and an operation manual for the man-machine interface to an experimental Omega Navigational Receiver. Included in the report are all circuit diagrams, system-interface connections, software flow charts, and software listings.

## 1.0 Introduction

To aid work being conducted at the Langley Research Center on the feasibility of a low cost Omega Navigational Receiver, a control panel was designed and constructed according to supplied specifications. Since the proposed Omega Receiver is designed around a microprocessor, software engineering necessary for control panel operation is also included in the design. The control panel is to be used as an operational model for use in the design of a prototype receiver.

The first part of this report provides a detailed description of the hardware design followed by a description of the software needed to operate the panel. The second part is a complete description of the operating procedures for the panel.

## 2.0 Control Panel Design

### 2.1 Specifications

The block diagram of the control panel is shown in Fig. 1. The following design specifications were used for the panel:

1. The control panel has two modes of operation: data entry (or change) and data display.
2. In the data entry mode, the operator may select one of eight parameters and input data regarding that parameter.
3. In the data display mode, the operator may display any one of eight input parameters or any one of eight calculated values.
4. The operator may change any input parameter at any time.
5. Certain indicators will be provided.

Most of these specifications involved both hardware and software considerations.

### 2.2 Hardware Implementation

The hardware problem was that of interfacing an INTEL 4004 4-bit microprocessor using a SIM-4 board to the receiver control panel. Some interface logic had been previously incorporated in the receiver such that the following input/output organization was dictated for the front panel interface:

1. One 4-bit data input bus (positive-logic)
2. One 4-bit data output bus (positive-logic)
3. Single bit control lines (negative-logic)

Most of the control lines were such that only one could be activated at any given time. However, a few of the control lines could be activated independently.

Details of the elements in the front panel block diagram of Fig. 1 will be explained in the following sections. Included on the front panel but not shown in the block diagram are a "power on" switch and a variable signal threshold selector (potentiometer). The threshold selector circuitry is shown in Fig. 2.

It should be noted that the choice of a large majority of the hardware used was purely on the basis of availability due to a limited time schedule. Therefore, it may be prudent to replace some of the components for the prototype version.

#### 2.2.1 Keyboard

Data entry was complicated by two constraints. The first constraint was the lack of interrupt capability in the INTEL 4004 microprocessor thereby forcing the computer to scan the keyboard, in some fashion, for a key closure. The second constraint was providing the capability to change the input parameters at any time. This meant that a parameter change could occur after the receiver had locked onto an Omega signal but keyboard scanning could occur only at given times. Also, not knowing how much time would be needed for calculations, it was not certain how much time would be available for the keyboard scan nor whether key closure could be easily detected.

Several schemes were investigated from a software key closure scan to a completely "buffered" keyboard. The software scan was a scheme in which the column of a keyboard matrix was enabled and the rows interrogated for a key closure. This scheme was abandoned because of the excessive computer time required since switch debouncing was done in software. The fully "buffered" keyboard scheme was one in

which the entire input parameter value would be held in flip-flop memory and the computer flagged only when this value was ready to be processed. This scheme required hardware to display and clear the characters, clock circuitry to shift the characters within the memory as new characters were entered, and a keyboard providing a coded output. This was abandoned due to excessive hardware cost.

The final design was a compromise of the above two schemes that incorporated the lower hardware cost of the first scheme and the minimum computer time of the second. The "semi-buffered" keyboard scheme allowed for storing one character until the computer could process it.

The circuit used is shown in Fig. 3. The keyboard, a Microswitch 12NW47-1, is a 4 x 3 coded keyboard with buttons for the digits 0-9, "enter", and "clear entry". The keyboard provides a negative-logic BCD code assignment and a delayed key closure strobe. When a button is depressed, the button code assignment is placed on the keyboard data lines and the strobe pulse, which is delayed for data line settling purposes, is generated. This strobe pulse sets a "data ready" flip flop (7472) which in turn activates a monostable multivibrator (74121). The multivibrator output is then used to load the keyboard data into a 4-bit latch, where it will remain until the computer can process it. The "data ready" flip flop (7472) will remain set until the computer generates a logic  $\phi$  on the "data ready reset" line. Until the "data ready" flip flop is reset, no new entry will be accepted and put into the data latch.

Information from the keyboard interface is entered into the computer via its 4-bit data input bus. The information is "wire-ored"



onto the bus by means of a tri-state device (80C97). The output of the "data ready" flip flop (7472) is placed on the bit  $\phi$  line of the data bus by a logic  $\phi$  on the "data ready enable" line, while the keyboard data is placed on the bus by a logic  $\phi$  on the "keyboard data enable" line. It is assumed that both enable lines and the "data reset" line are logic "1" normally (positive logic).

It has been shown, therefore, that with this scheme the computer has only to process keyboard information if a key has been depressed, thereby greatly reducing the computer time needed. In addition, an entry will not be lost if the computer is not able to accept it at time of entry.

#### 2.2.2 Display

The display circuit, shown in Fig. 4, is common to many displays. The display consists of four digits using seven segment light-emitting diodes (LED). Each digit is a Fairchild FND5 $\phi\phi$  LED and uses a Fairchild matching seven-segment driver (9368) in order to eliminate the need for current-matching resistors. The LED drivers include both a 4-bit latch and a BCD-to-seven segment decoder. This allows the display data to be put on the computer's data output bus and loaded into the correct display digit latch using a control line from the computer. The decoder/driver also includes the ripple blanking capability which was used for the display. The display control lines were normally at logic level "1" while a logic " $\phi$ " enables the corresponding digit latch.

#### 2.2.3 Data Selector Circuit

The data selector circuit allows the operator to select one of eight variables to be entered or displayed or one of eight results to be displayed. Since the number of result values equals the number

of input variables, the circuit is designed so that each position of the data selector will indicate either an input variable or a result value. Which of these two parameters is chosen is determined by a mode switch.

The data selector circuit is shown in Fig. 5. The data selector is a double-pole, non-shorting, eleven position rotary switch of which only eight positions are used. A 5 volt, 60 milliamperes incandescent lamp, used to indicate switch position, is connected to each of the eight positions used on one pole (wafer) of the switch. The contacts of the eight positions on the other pole (wafer) of the switch are connected to the eight least significant output lines of a four-line-to-sixteen-line decoder (7442). The wiper of this pole is connected to the computer data input bus via a tri-state device (80C97). The inputs of the decoder (7442) are connected to the output data bus of the computer. The rotary switch position information is obtained by placing a BCD number between zero and eight on the output data bus. This will cause one of the outputs of the decoder (which are normally at logic "1") to obtain a logic " $\phi$ ". If the wiper of the rotary switch is on that decoder output, the logic " $\phi$ " is placed on the input bus. The computer varies the value on the output data bus until a logic " $\phi$ " is detected on data line  $\phi$  of the input bus which indicates the switch position has been found. The value on the output bus is used as an indication of the actual switch position. A pull-up resistor (R4) is necessary on the rotary switch wiper to maintain a logic "1" while the switch is between positions.

The data mode switch is a double-pole, double-throw, center-off toggle switch. It is used to indicate one of three modes; data entry,

entry data display, and result data display. This switch provides a code to be placed on lines 1 and 2 of the data input bus from which the computer can determine the correct mode of operation. The switch also activates a red incandescent lamp, labeled "data in", when in the data entry mode. All of the data selector circuit information is placed on the computer's data input bus by a logic " $\phi$ " on the "rotary switch enable" line. The codes for the data selector positions and mode switch positions are given in Fig. 6.

#### 2.2.4 Indicator Circuits

Several computer controlled indicator lamps are provided on the front panel. These lamps include indications for distance to destination less than five miles, signal sync detection, and arrows for track deviations. The circuits for these indicators are shown in Fig. 7. The indicators are 5 volt, 60 milliamperes incandescent lamps driven by DTL (diode-transistor logic) open-collector power gates (858). These particular power gates were necessary because of current requirements of the lamps. The two input nand gates (7400), used as inverters, enable a logic " $\phi$ " from the computer to activate the lamps.

#### 2.3 Hardware Results

A picture of the experimental control panel is shown in Fig. 8. The dimensions of seven inches by ten inches were forced by the hardware used. It should again be noted that the criteria used for choosing the particular hardware was mainly availability. This fact did force the panel to be larger than necessary and can be reduced in size in the prototype version.

The positions of the components on the two printed circuit boards used are shown in Figs. 9 and 10. The dimensions of these printed

circuit boards are actual size in the figures. All interfacing to the front panel is done through a 36 pin connector. Fig. 11 gives these front panel pin connections.

## 2.4 Software Implementation

An overview of the Omega Receiver software requirements is shown in Fig. 12. The input/output pilot interface or front panel operation software is a completely self-contained subroutine package (except for two system subroutines) which requires no parameter transfer upon entry. The two subroutines which are not included in the software package operate the control lines used by both the pilot interface and the Omega data acquisition circuits. One of the subroutines, AENAB, produces a pulse output on a control line specified upon entry. The other subroutine, BENAB, produces a level output on a control line again specified upon entry. The front panel software package contains a main routine and nine subroutines. A detailed explanation of these programs follows.

The flow charts for the software package is found in Appendix A. Appendix B contains the program listings.

### 2.4.1 Main Routine

The main routine of the front panel control software is organized as a subroutine which needs no parameters upon entry. Operation of the front panel is solely by a call to this subroutine. For each entry to this main routine, only one panel operation is performed, i.e. the displaying of a value or the processing of a character that has been entered.

Upon entry to the routine, the computer must first determine by interrogating the mode switch whether the operation requested is data

entry or data display. If data display is requested the position of the data selector and mode switches are then determined. The position of the mode switch is needed to determine if the data of interest is input parameter data or calculated result data. Since all data values are stored in binary format in the computer, the value of interest is converted to BCD (Binary coded decimal) format which is needed by the display unit. The value is displayed on the LED display and the panel routine is exited.

If, on the other hand, it is found that the operation requested is data entry, the panel is interrogated for a character awaiting entry. If this is the first data entry request since a data display request, four RAM (random access memory) locations are zeroed and LED display unit is blanked. The four memory locations are used as a buffer to hold the entered digits until an "E" or "CE" command is entered. The routine is exited if no keyboard entry is found. If, however, a keyboard entry is available, the computer inputs the entry and decides whether it is an "E", a "CE", or a digit. For a digit entry, each digit already in the memory buffer is shifted to its next higher significant position and the new entry then becomes the least significant digit. The most significant digit of the digit buffer area is always lost. The contents of the buffer area is displayed and the routine is exited.

A "CE" (clear entry) causes the digit buffer area to be zeroed and the display to be blanked. If the entry is an "E", the contents of the digit buffer, which is in BCD format, is converted to binary format and stored in three memory locations in the computer. The specific memory locations are determined by the position of the data selector. The digit buffer area is then zeroed, the display blanked, and the routine exited.

#### 2.4.2 Display Subroutine

The Display subroutine, OUTSB, displays the four BCD digits found in the RAM digit buffer area onto the LED seven-segment display. No parameters are transferred upon entry. The displaying is accomplished by consecutively placing one of the BCD digits on the output data bus and activating its control line.

#### 2.4.3 Rotary Switch Position Subroutine

The Rotary Switch Position subroutine, ROTPOS, determines the position of the data selector switch. The data output bus address, the data input bus address, and the data selector enable code are transferred into the routine when the subroutine is called. A register returns from the routine with an index number between 0 and 7 indicating the switch position.

#### 2.4.4 Rotary Switch Address Subroutine

The Rotary Switch Address subroutine, ROTSW, determines the address of a specified binary data value. The routine requires, upon entry, that the starting address of the binary data value area be contained in one register pair and an index (0-7) in another register. Upon return from the subroutine, the register pair will have been modified to contain the address of the least significant bits of the specified data value. The subroutine assumes that each data value occupies three 4-bit bytes.

#### 2.4.5 Addition Subroutine

The addition of two 12-bit numbers is performed by the Addition subroutine. The subroutine has two entry points, ADD and ADD1. Entering the routine at ADD resets the carry condition bit to zero while entering at ADD1 leaves the carry bit unaffected. The two 12-bit numbers are

transferred into the subroutine in two groups of three index registers each. The result of the addition returns from the subroutine in one of these groups of index registers.

#### 2.4.6 Subtraction Subroutine

The Subtraction subroutine performs the subtraction of two 12-bit numbers. This subroutine also has two entry points, SUB and SUB1, which provide the same capability as in the Addition subroutine; that is, resetting the carry bit. The transfer of information to and from the Subtraction subroutine is accomplished as in the Addition subroutine.

#### 2.4.7 Increment Subroutine

The Increment subroutine is used to increment the contents of a particular register pair. The subroutine has two entry points, INC and INC1. Entering the routine at INC1 will cause the register pair to be incremented by one. Entering at INC, however, will cause the register pair to be incremented by the contents of the accumulator. The particular register pair is fixed within the subroutine and is not transferred into the routine as a parameter.

#### 2.4.8 Zero Ram Subroutine

The Zero Ram subroutine, with entry point BLANK, is used to zero four consecutive RAM locations. The starting RAM address is transferred to the routine in a register pair. A limitation to the routine occurs since the consecutive RAM addresses are determined by incrementing only the four least significant bits and not the entire eight bit address. This means that the four most significant bits never change regardless of the starting RAM address. Details and ramifications of this can be easily seen by examining the program listing.

#### 2.4.9 Multiply Subroutine

The Multiply subroutine, MUL, is used to multiply a twelve bit number by a four bit number. The result of this multiplication must not exceed twelve bits. Upon entry to the subroutine, two registers and the accumulator will contain the twelve bit number and the address of the four bit number will be found in a register pair. On exit, the result is placed in a group of three index registers. The multiplication is accomplished within the routine by repetitive addition. The main use of this subroutine is for BCD to binary conversion.

#### 2.4.10 Divide Subroutine

The Divide subroutine, DIV, performs the division of two twelve bit numbers. The result of this operation can occupy no more than four bits or a memory byte. Two groups of index registers, three registers each, contain the two numbers at time of entry to the routine. Another register pair contains the address into which the result will go. At exit from the routine, one of the index register groups contains the remainder and the register pair contents have been decremented. The division operation is accomplished by repetitive subtraction. This subroutine helps perform binary to BCD code conversion.

#### 2.5 Software Results

As stated before, the control panel software package requires no additional software other than the two system subroutines mentioned. Also, no additional software is needed during power-up initialization phase. No other program, however, must alter the contents of certain RAM locations. These locations are those specified by hexadecimal addresses 2 $\phi$  through 23 and RAM status character  $\phi$  of RAM  $\phi$ . The



software requires 288 words of PROM (programmable read-only memory). Optimization of the software with respect to minimizing memory used has been attempted; however, still more optimization may be possible.

### 3.0 Operating Procedures

The operating procedure for the experimental control panel will be presented in a step-by-step manner. The nomenclature will refer to that on the constructed panel which is shown in Fig. 8.

#### 3.1 Data Input

1. Position the mode selector switch labeled "DATA" to the "IN" position. The "DATA IN" lamp will light.
2. Position the rotary data selector switch to the desired variable. The labels above the green lamps are the input variables. The appropriate green lamp will light.
3. Enter the desired number followed by a "E" (#). As each digit is entered it will appear on the display. Only the last four digits entered are significant. A "CE" (\*) entry will remove the digits previously entered thereby blanking the display. A "E" (#) entry will assign the number in the display to the parameter chosen by the data selector.

NOTE: The above procedure is used whether initially entering the parameters or at any time changing a parameter or parameters.

#### 3.2 Data Display

1. Position the mode selector switch labeled "DATA" to the center position for displaying input parameters or to the position labeled "OUT" for the calculated results.
2. Position the rotary data selector switch to the desired parameter. The labels above the lamps are the input parameters and the labels below are the calculated results. The appropriate lamp will light and the value of the parameter will be displayed.

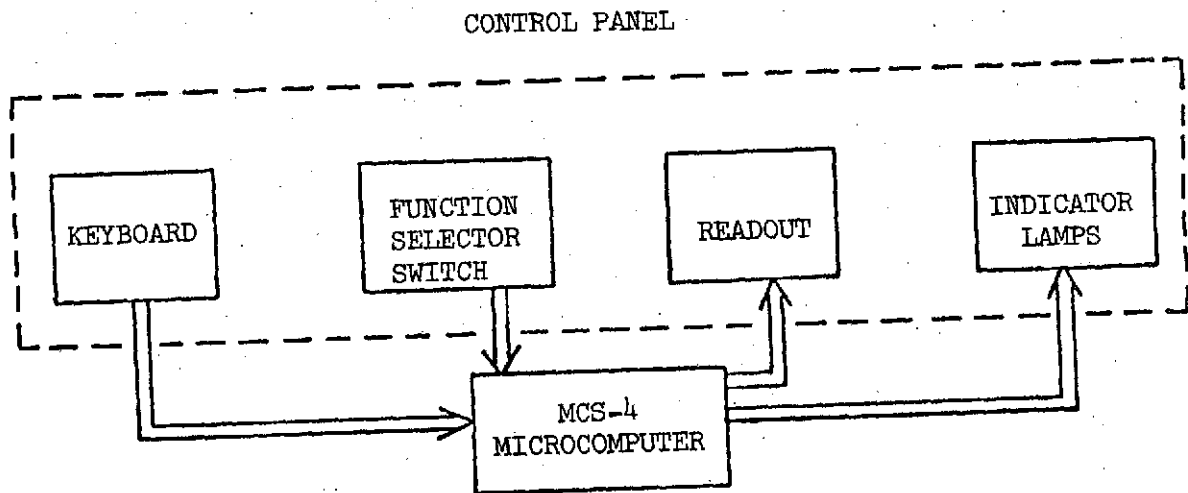


Figure 1. Control Panel Block Diagram

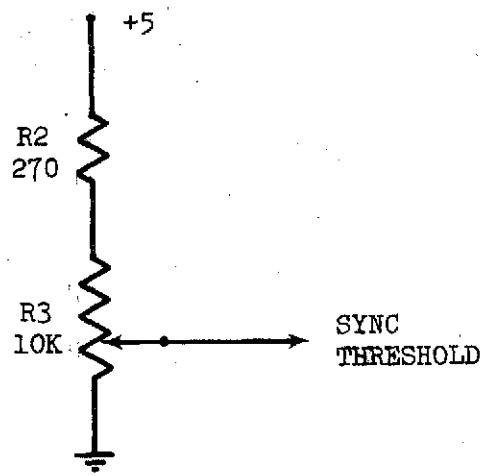


Figure 2. SYNC Threshold Adjustment

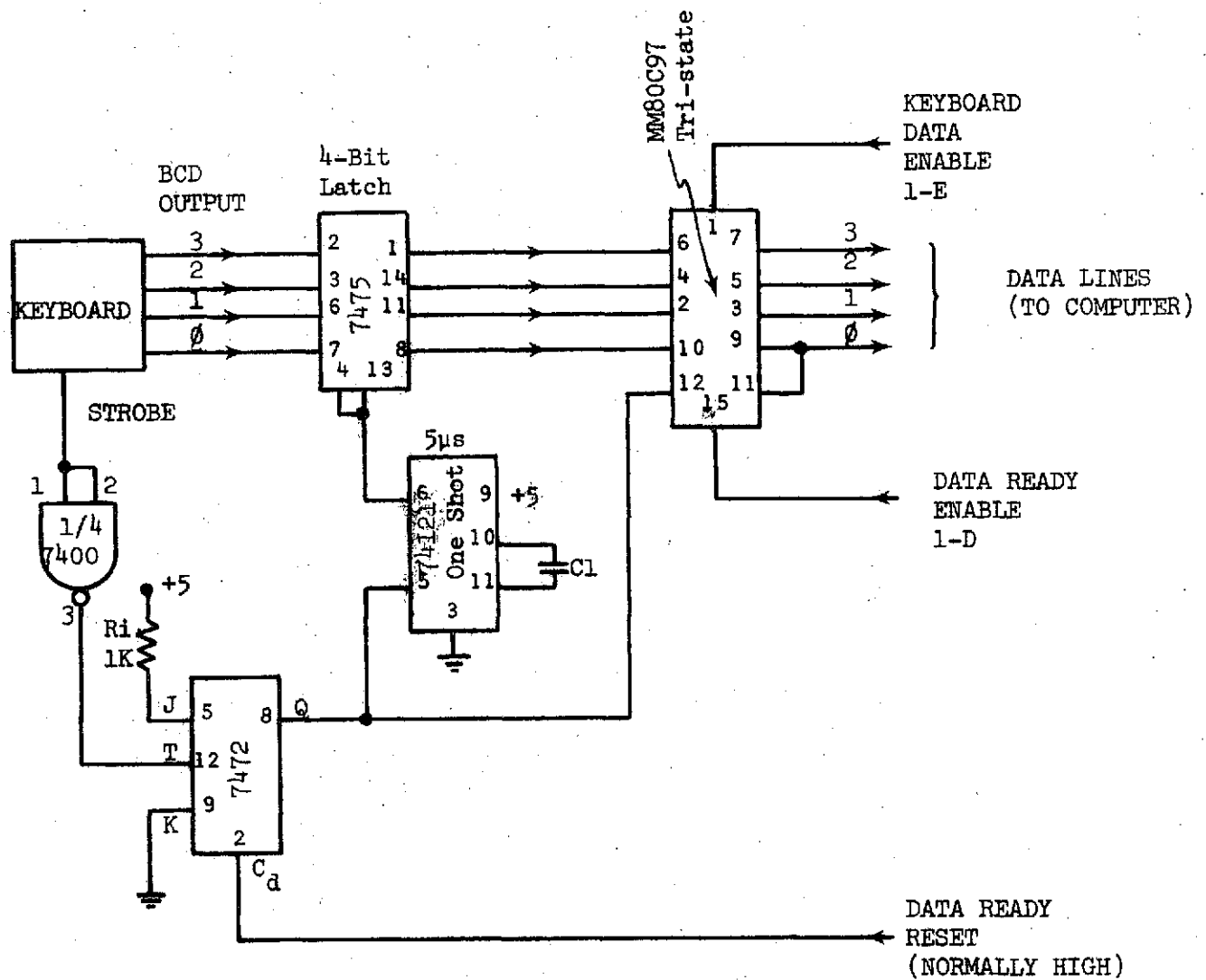


Figure 3. Coded Keyboard Interface

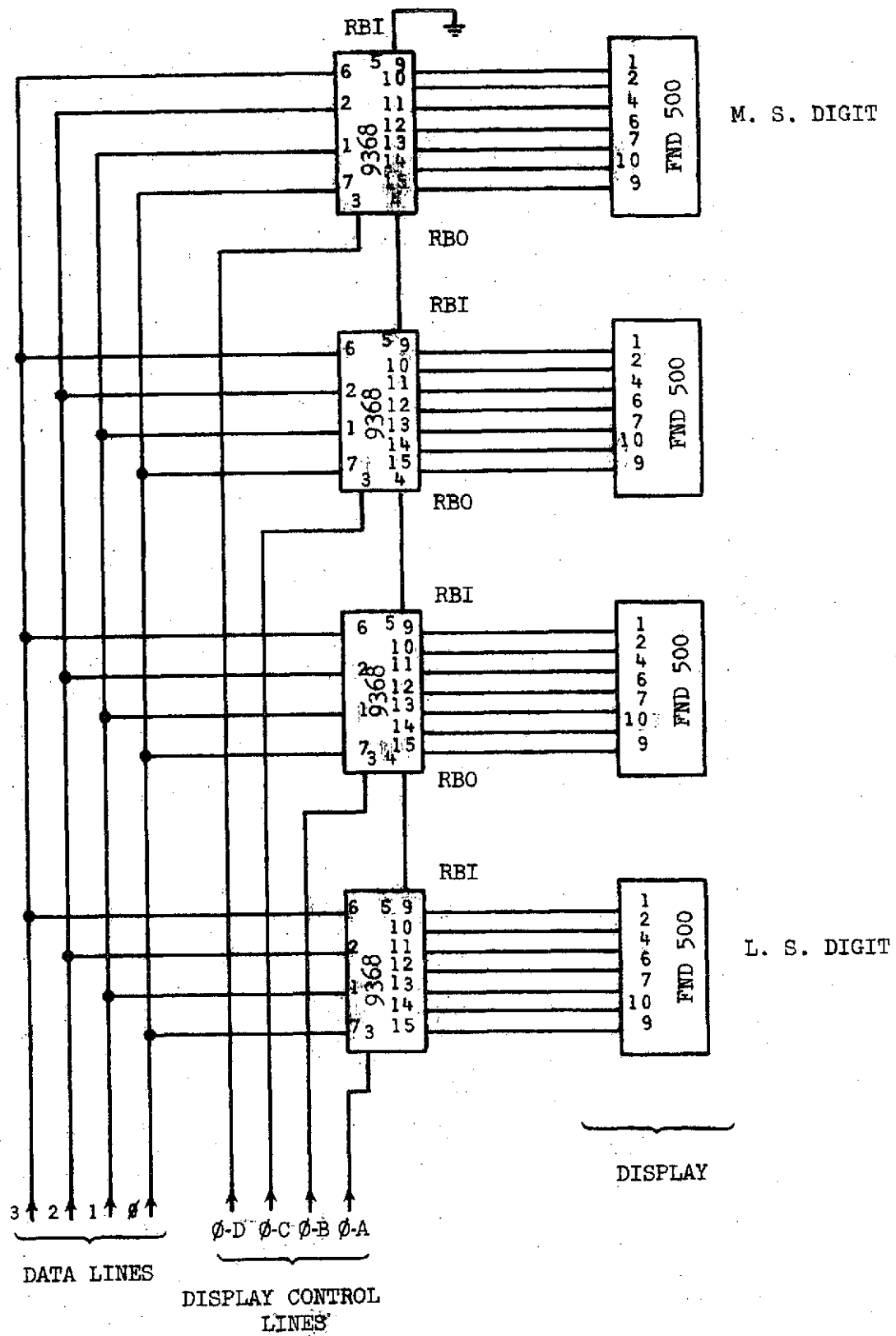


Figure 4. Display Interface

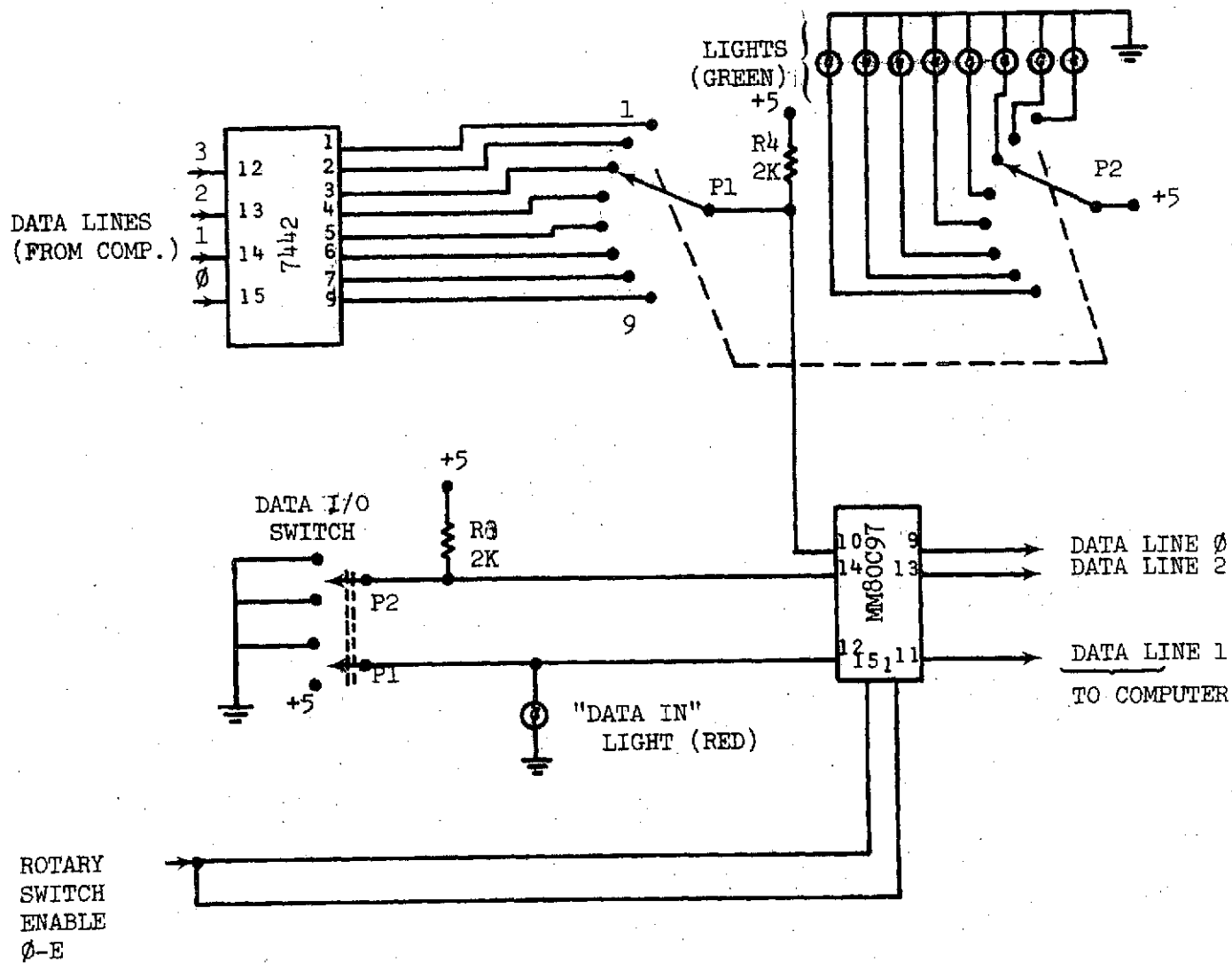


Figure 5. Data Selector Interface

DATA SELECTOR CODE

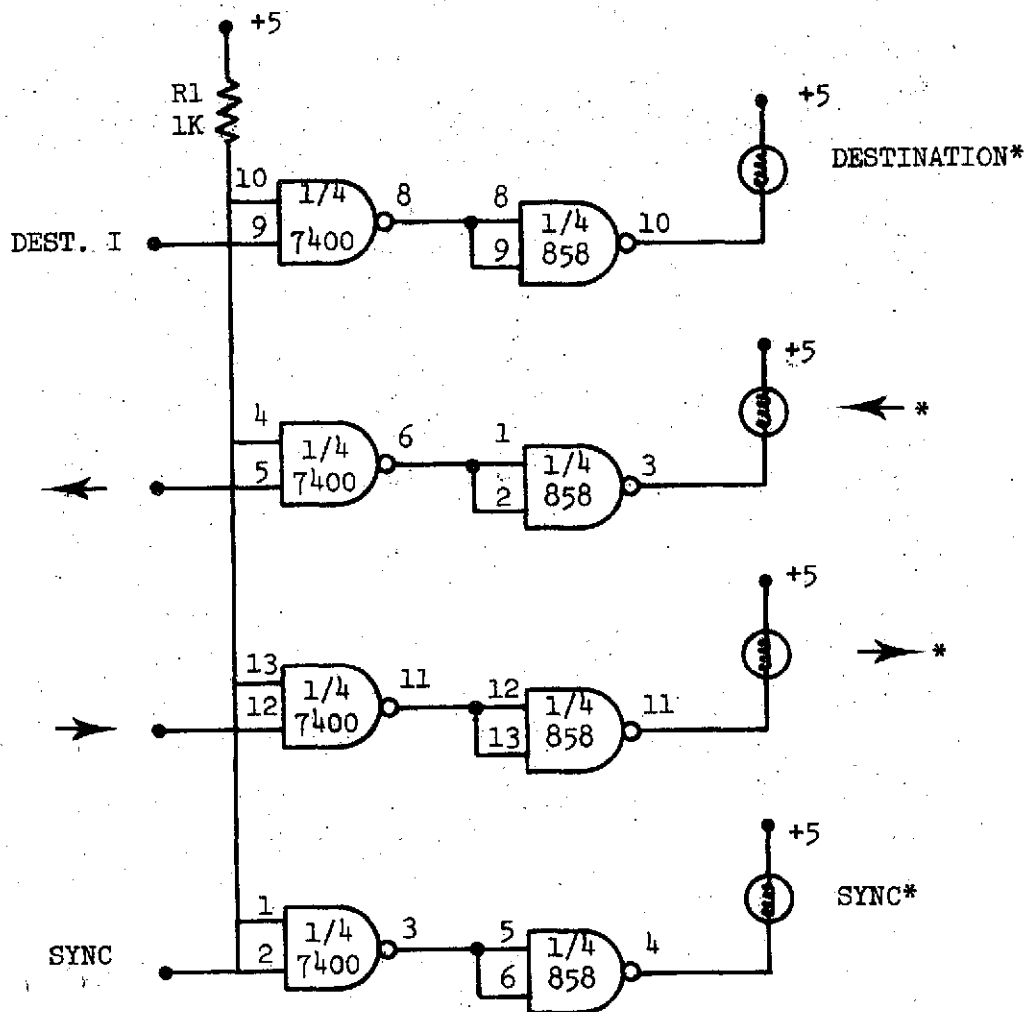
CODE (BINARY)	SELECTOR LABEL
000	X DESTINATION, X POSITION
001	Y DESTINATION, Y POSITION
010	X SET, HEADING CORRECTION
011	Y SET, BEARING TO DESTINATION
100	$\phi_x$ , DISTANCE TO DESTINATION
101	$\phi_y$ , TIME TO DESTINATION
110	$D_x$ , GROUND SPEED
111	$D_y$ , TRACK HEADING

MODE SELECTOR CODE

CODE (BINARY)	SELECTOR LABEL
00	DATA RESULT DISPLAY (DOWN POSITION)
01	DATA INPUT (UP POSITION)
10	DATA INPUT DISPLAY (CENTER POSITION)

Figure 6 Selector Codes





\*5 volt, 60 milliamperes incandescent lamp

Figure 7. Indicator Circuits

ORIGINAL PAGE IS  
OF 2001-001111



Figure 8. Picture of Final Product

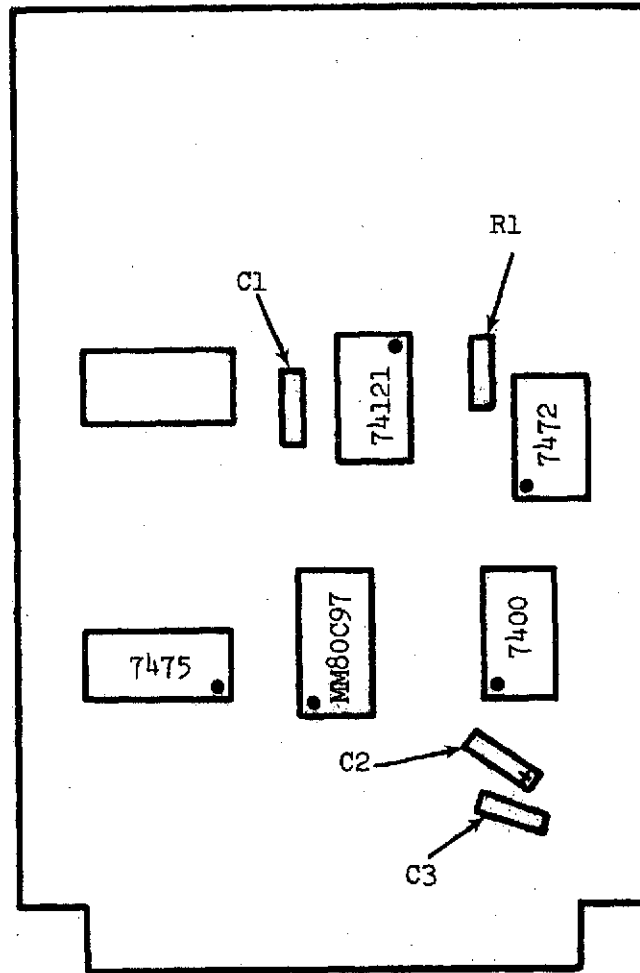


Figure 9. Keyboard Interface  
Bottom View  
(Circuit Side)(Actual Size)

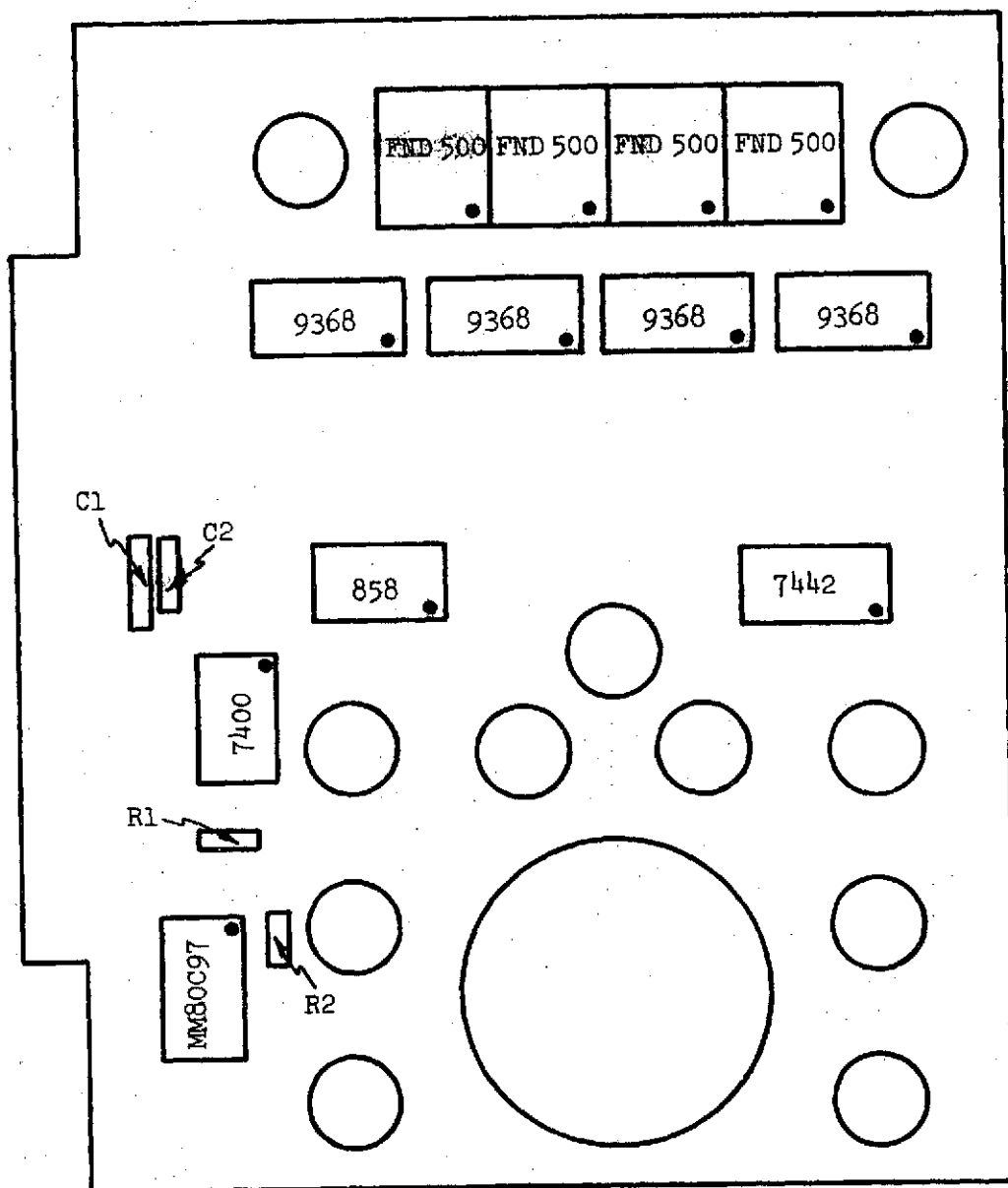


Figure 10. Front Panel Interface  
Bottom View  
(Circuit Side)(Actual Size)

<u>PIN #</u>		<u>F.P. CONN.</u>	<u>PIN #</u>		<u>F.P. CONN.</u>
1	-	+5 VOLTS	19	-	+5 VOLTS
2	-	NC	20	-	NC
3	-	BIT $\phi$ }	21	-	BIT $\phi$ }
4	-	BIT 1 } DATA INPUT	22	-	BIT 1 } DATA INPUT
5	-	BIT 2 } BUS	23	-	BIT 2 } BUS
6	-	BIT 3 }	24	-	BIT 3 }
7	-	KB DATA ENABLE	25	-	READOUT ENABLE(L.S.DIGIT)
8	-	DATA READY ENABLE	26	-	READOUT ENABLE
9	-	KB RESET	27	-	READOUT ENABLE
10	-	CENTER TAP OF POT	28	-	READOUT ENABLE(M.S.DIGIT)
11	-	NC	29	-	ROTARY SWITCH ENABLE
12	-	NC	30	-	NC
13	-	NC	31	-	DESTINATION LIGHT
14	-	NC	32	-	← LIGHT
15	-	NC	33	-	→ LIGHT
16	-	NC	34	-	SYNC LIGHT
17	-	NC	35	-	NC
18	-	GROUND	36	-	GROUND

Figure 11. Front Panel Pin Connections

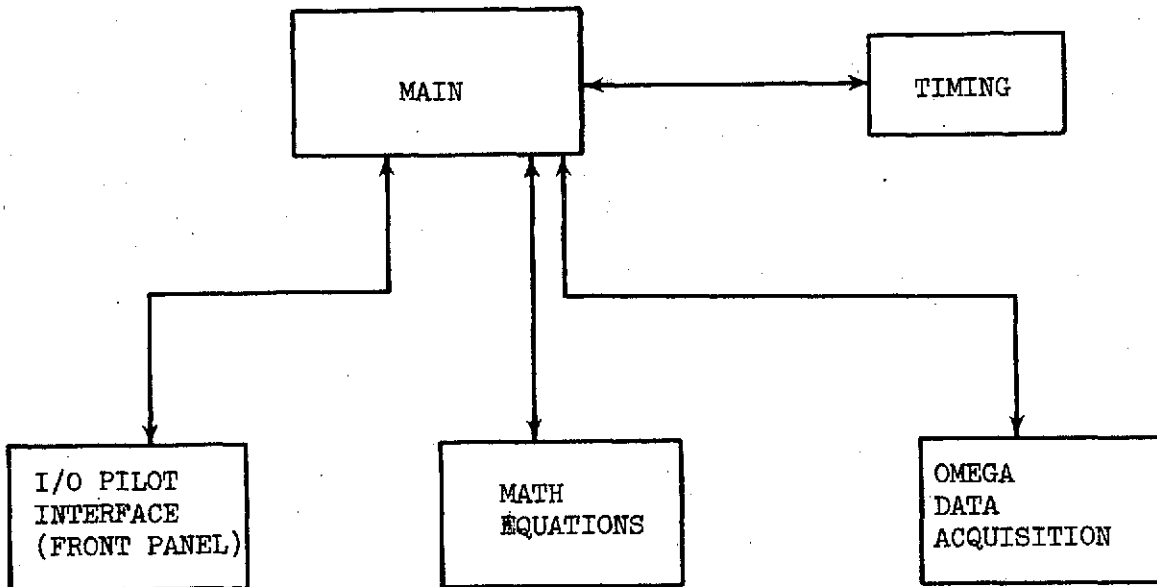
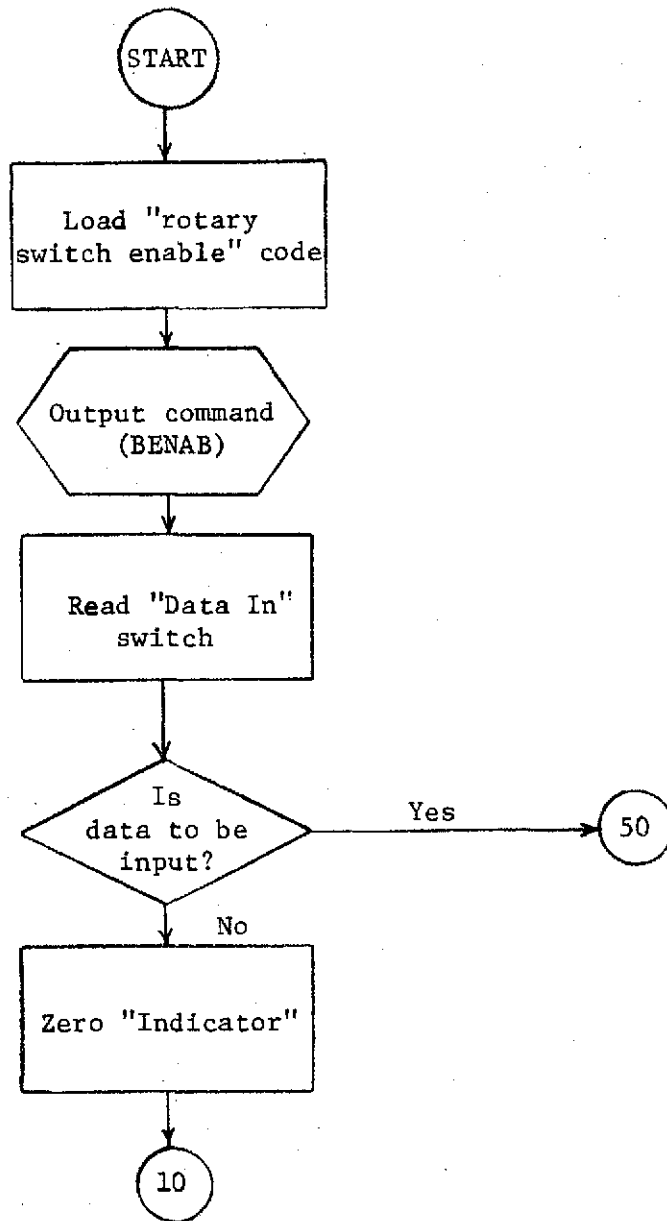


Figure 12. Omega Receiver Software Requirements

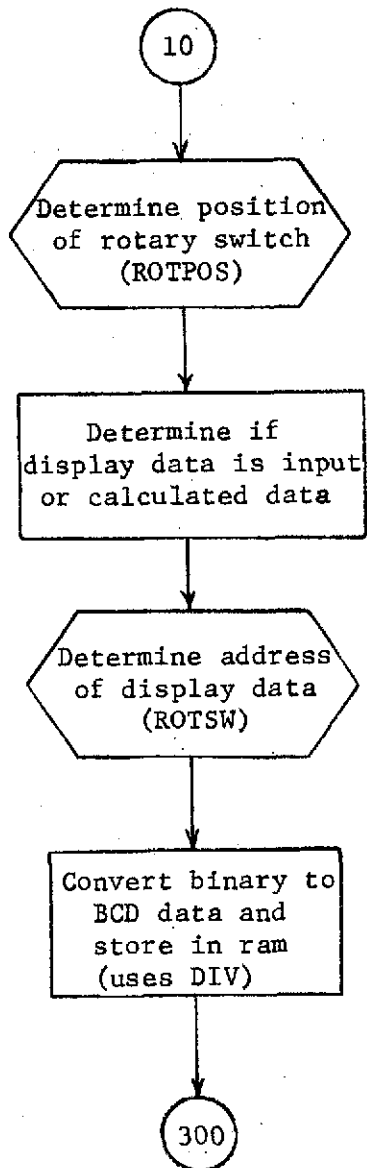
APPENDIX A

FRONT PANEL MAIN PROGRAM

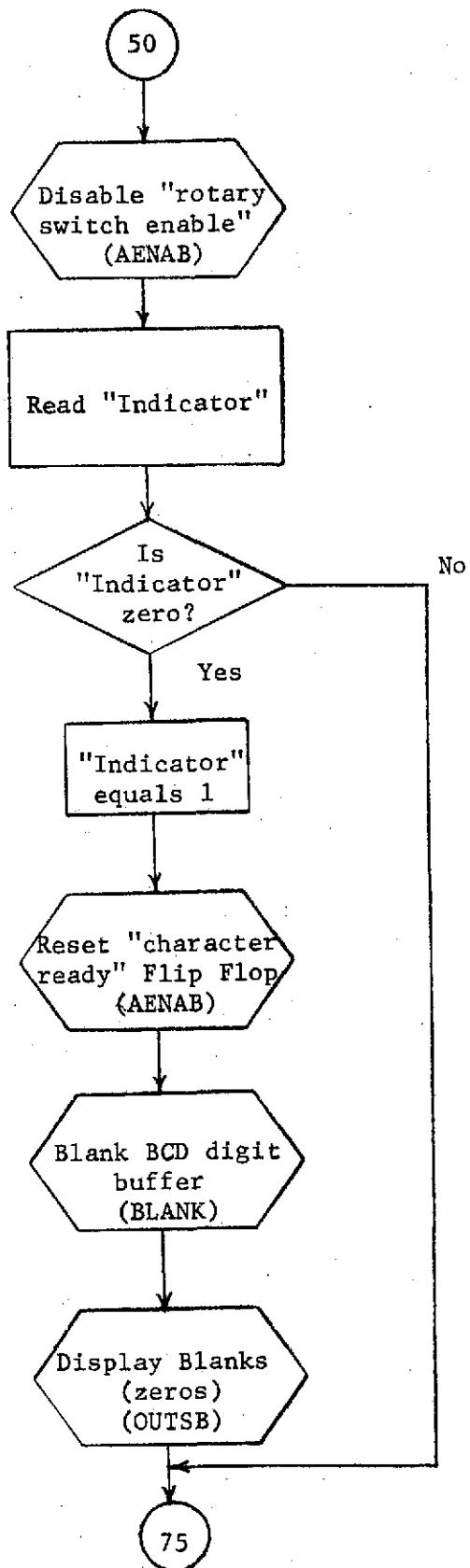




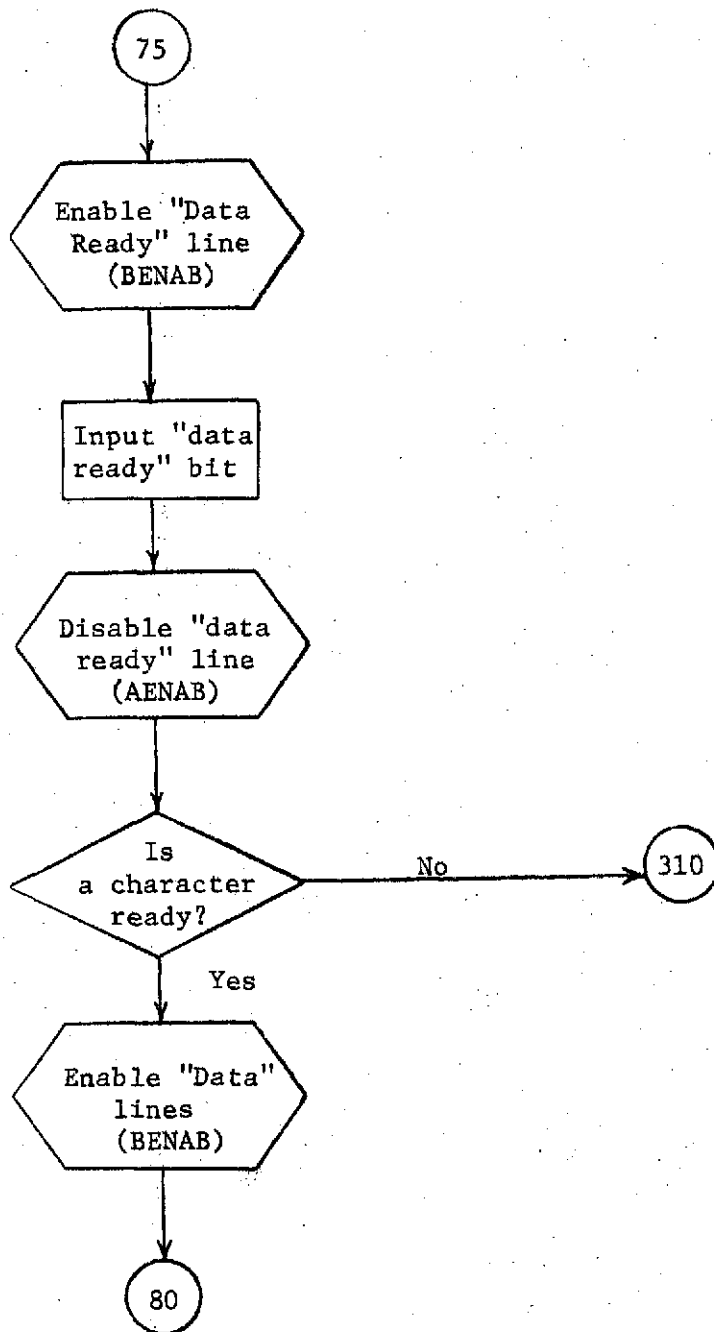
FRONT PANEL MAIN PROGRAM CONTINUED



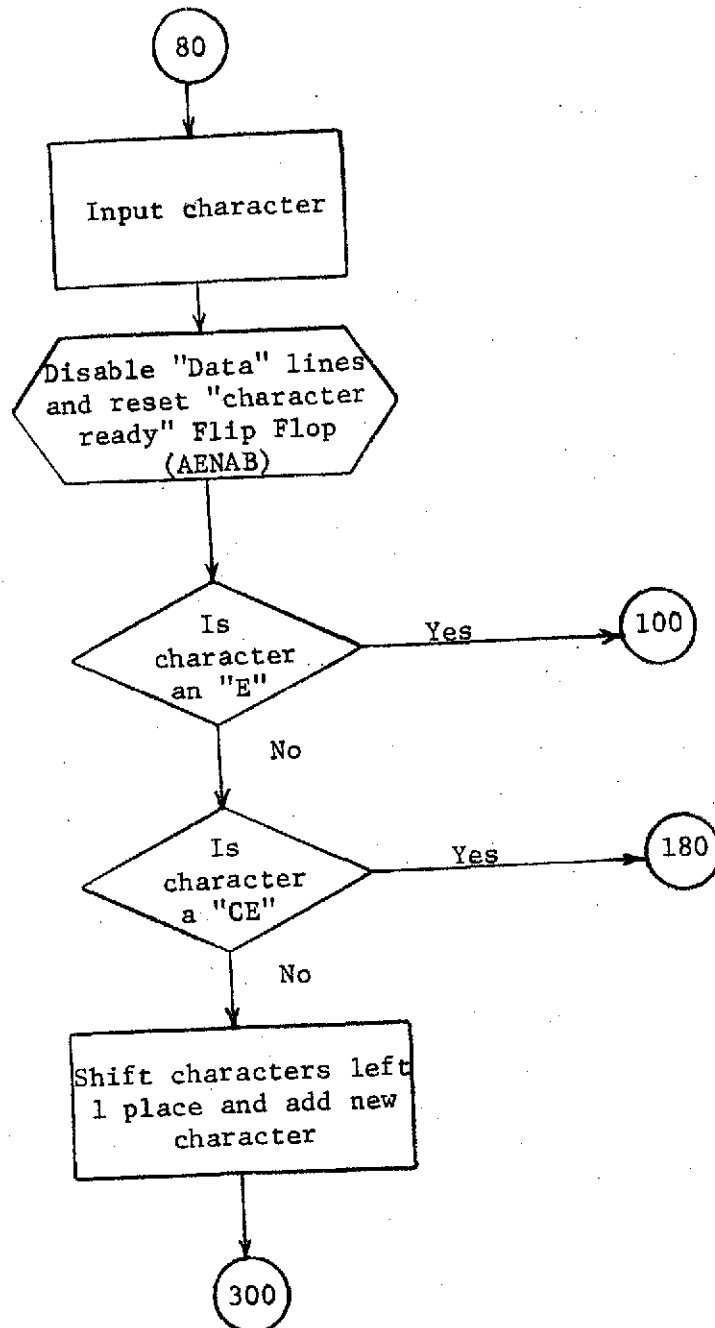
FRONT PANEL MAIN PROGRAM CONTINUED



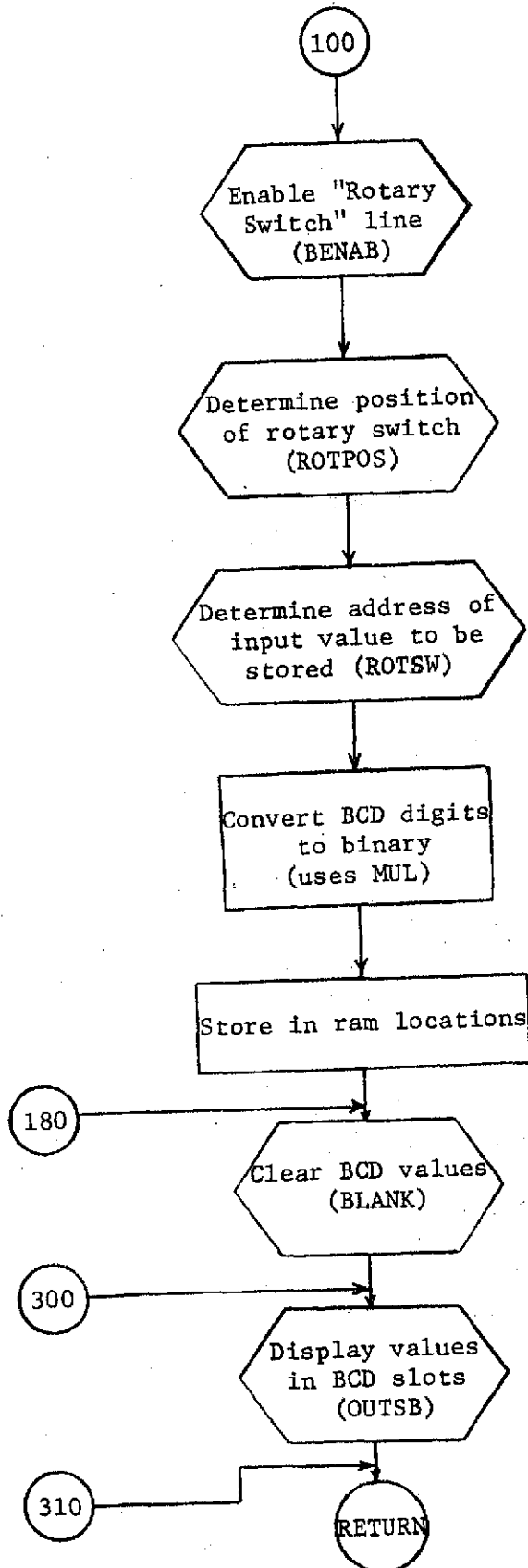
FRONT PANEL MAIN PROGRAM CONTINUED



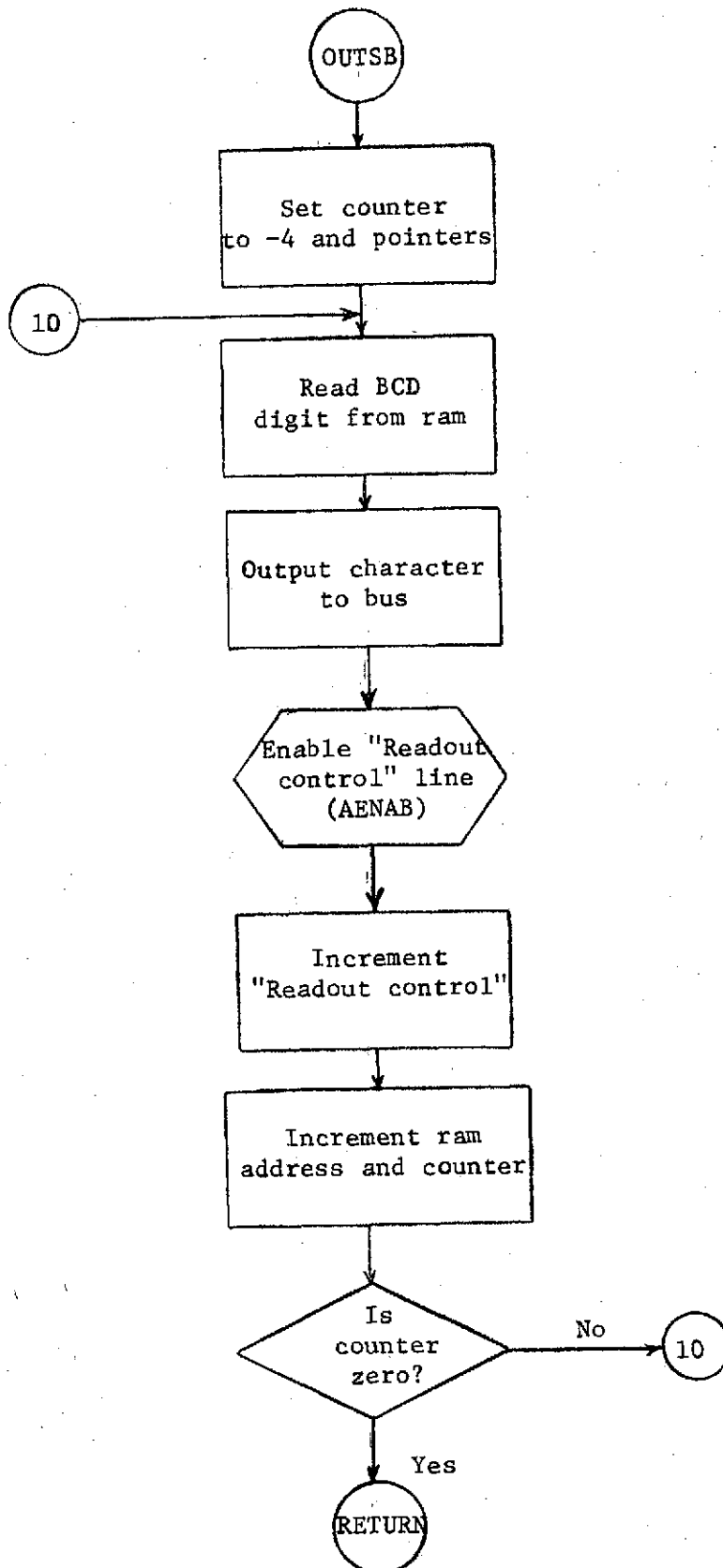
FRONT PANEL MAIN PROGRAM CONTINUED



FRONT PANEL MAIN PROGRAM CONTINUED



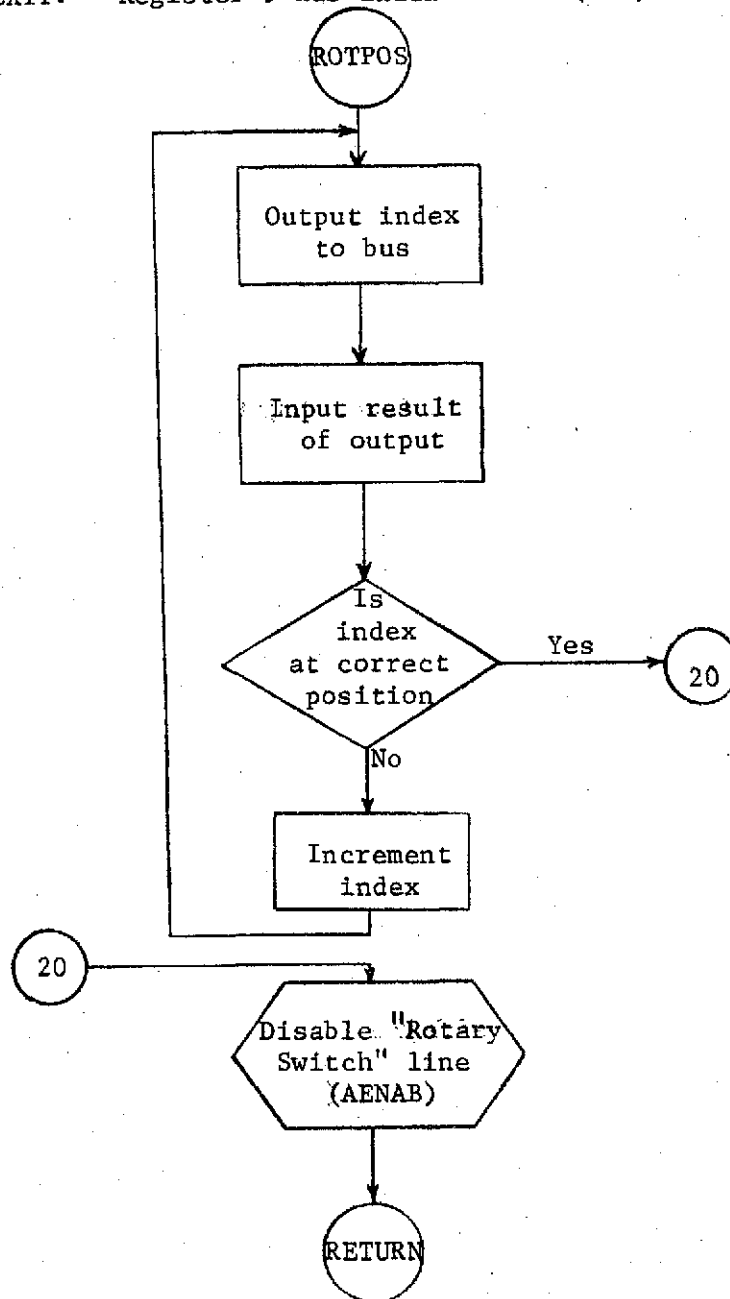
# DISPLAY OUTPUT SUBROUTINE



# ROTARY SWITCH POSITION SUBROUTINE

ON ENTRY: Register pair 5 has output port address.  
Register pair 4 has input port address and  
the starting index number.  
Register pair 6 has the rotary switch enable code.

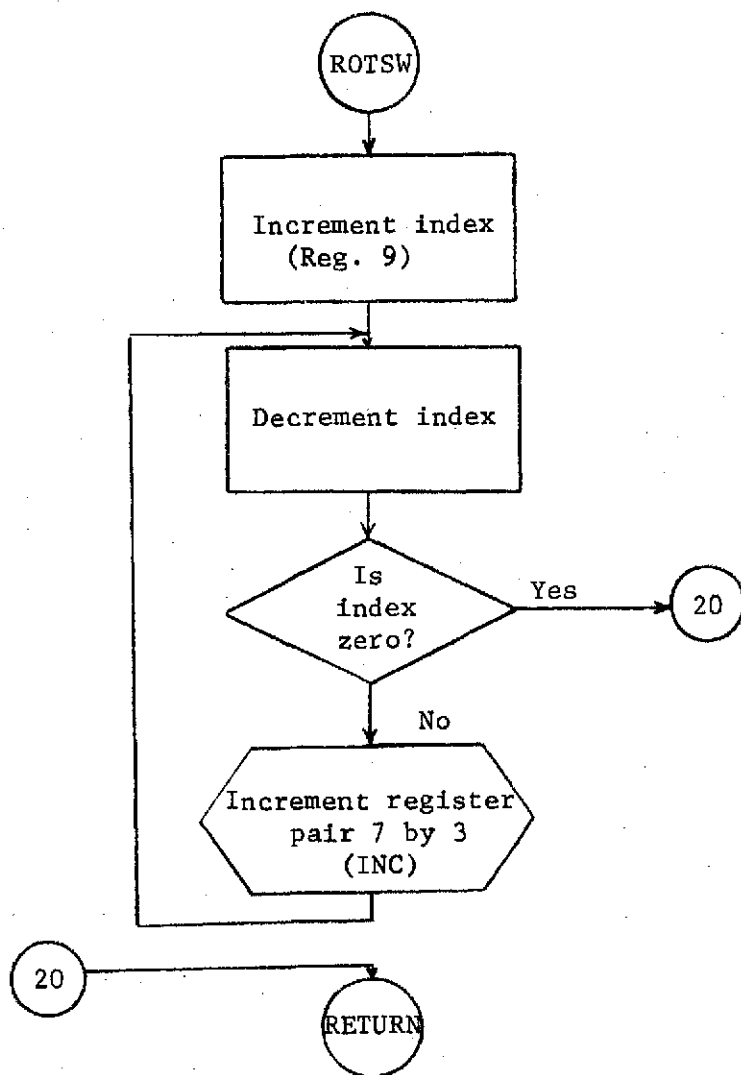
ON EXIT: Register 9 has index number (0-7).



# ROTARY SWITCH ADDRESS SUBROUTINE

ON ENTRY: Register 9 contains index position of rotary switch (0-7)  
Register pair 7 has base address of data area

ON EXIT: Register pair 7 has the address of the least significant bits of the binary data of interest.

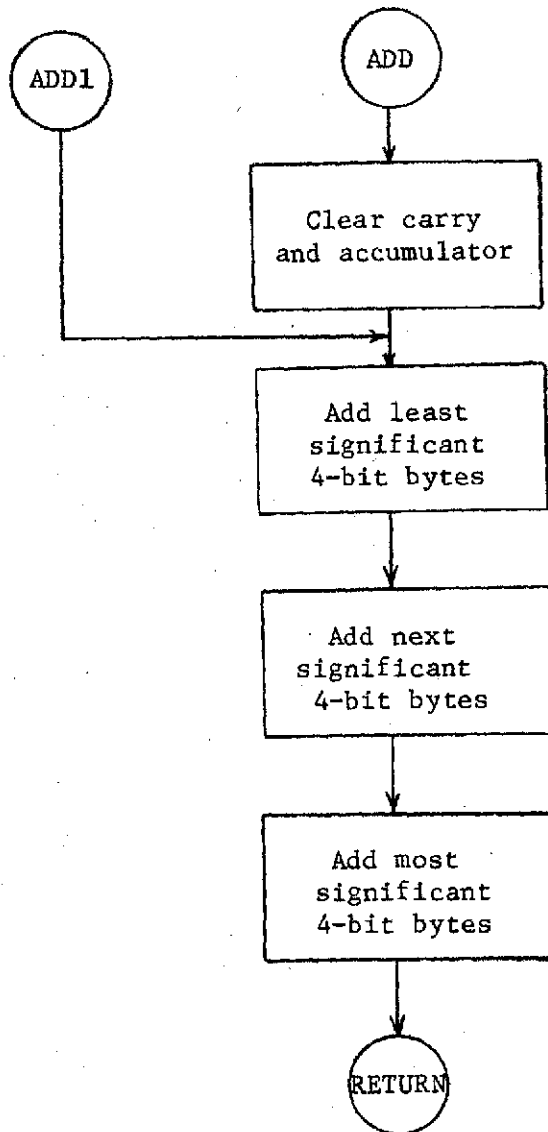




### ADDITION SUBROUTINE

ON ENTRY: Registers 6, 7, 8 have a 12 bit number  
Registers 9, 10, 11 have a 12 bit number

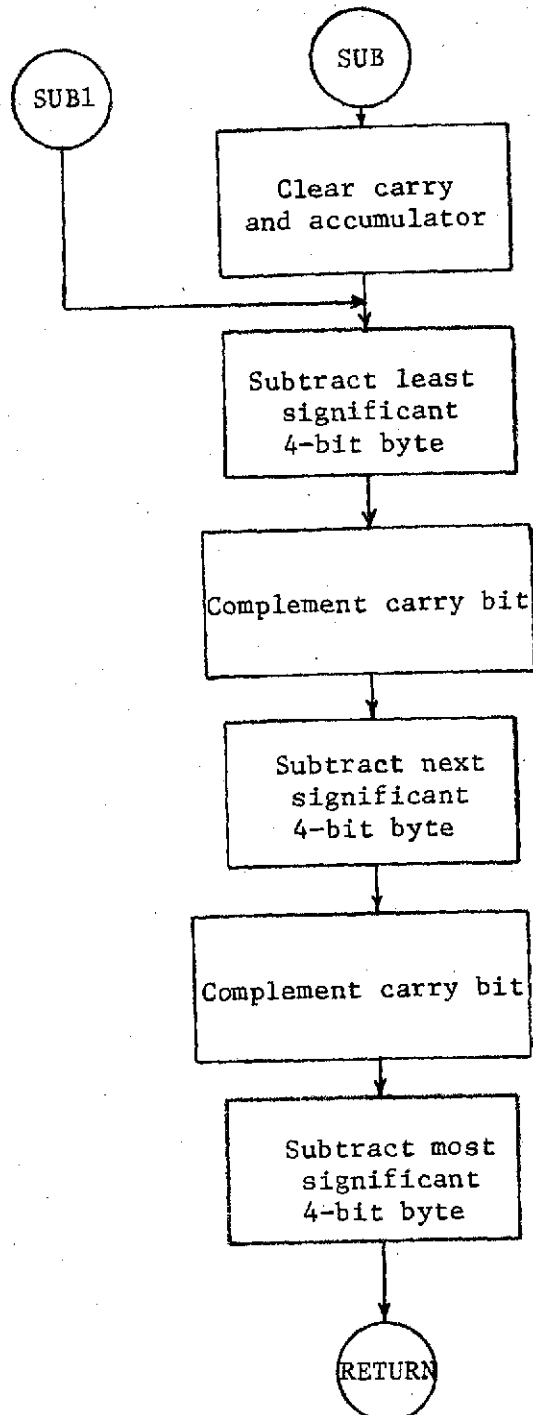
ON EXIT: Register 9, 10, 11 have the result of the sum  
of the two 12 bit numbers



### SUBTRACTION SUBROUTINE

ON ENTRY: Registers 6, 7, 8 contain a 12 bit number  
Registers 9, 10, 11 contain a 12 bit number

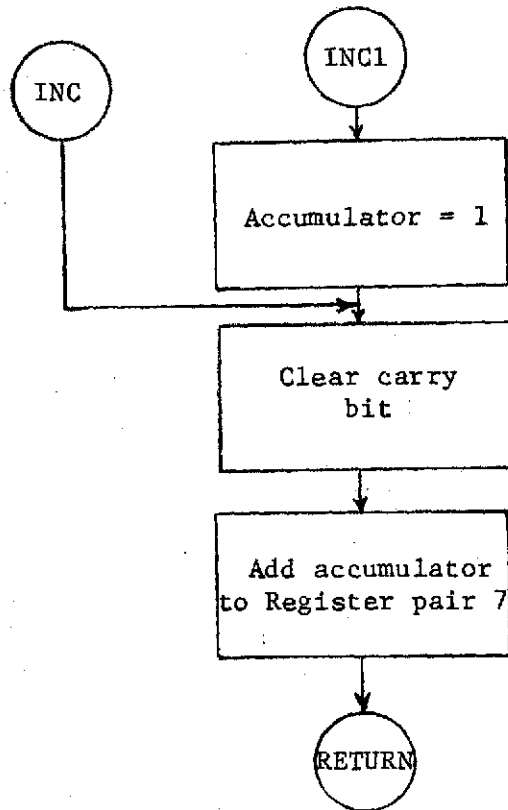
ON EXIT: Registers 9, 10, 11 contain the 12 bit result



### INCREMENT SUBROUTINE

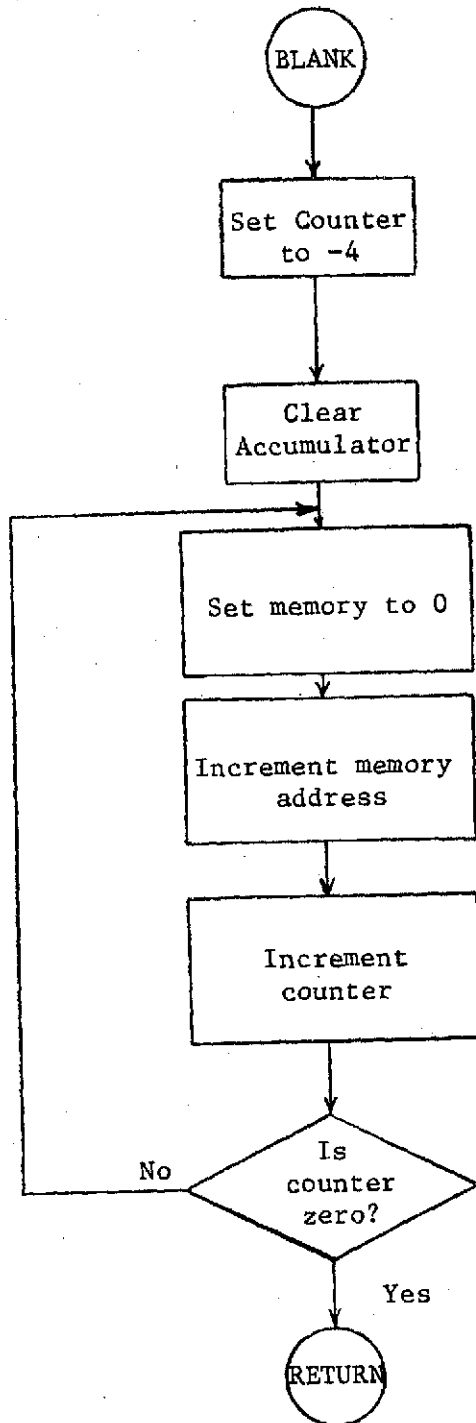
ON ENTRY: Register pair 7 has a value  
Accumulator may have a value

ON EXIT: Register pair 7 has a modified value



# ZERO RAM SUBROUTINE

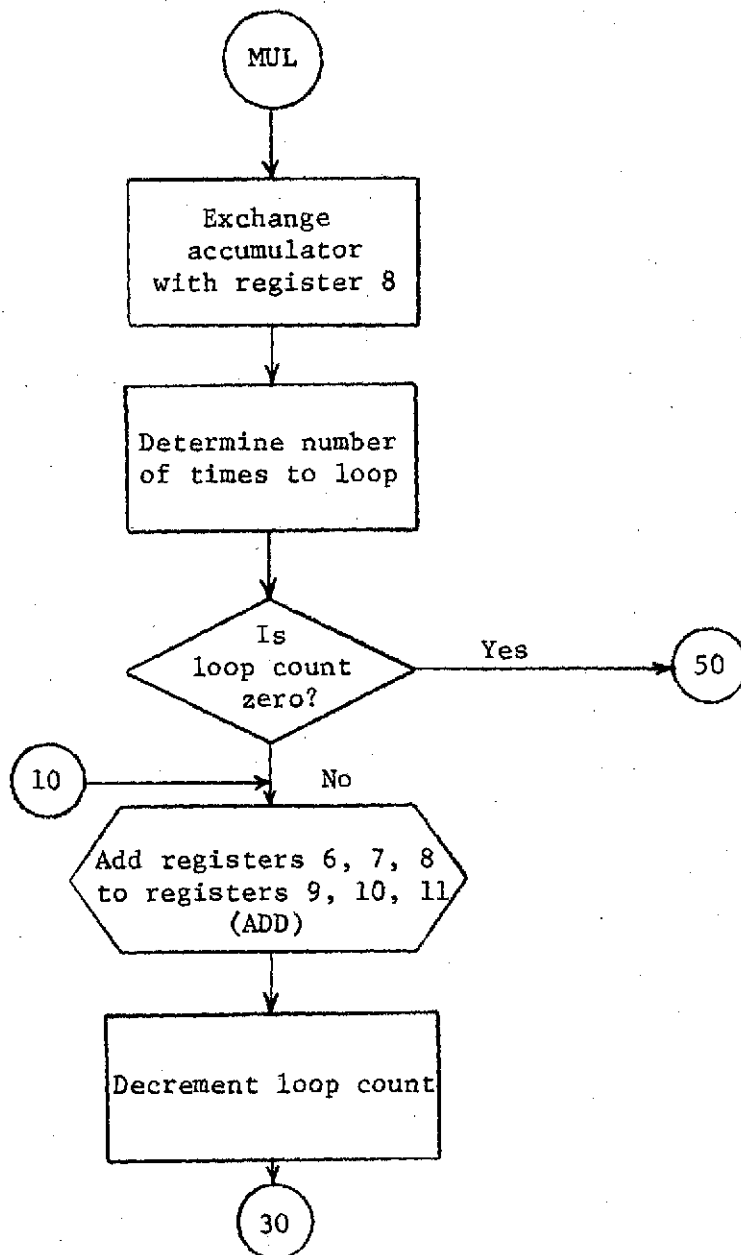
ON ENTRY: Register pair 5 has a ram address



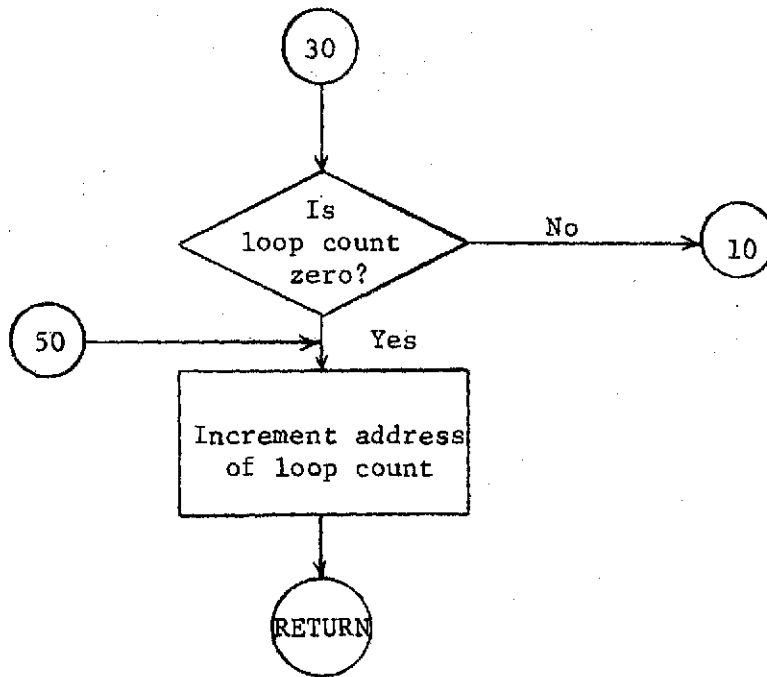
### MULTIPLY SUBROUTINE

ON ENTRY: Registers 6, 7, accumulator contain one multiplier  
Register pair 6 contains the address of the other multiplier

ON EXIT: Binary result is added to registers 9, 10, 11  
Register pair 6 contains the next address



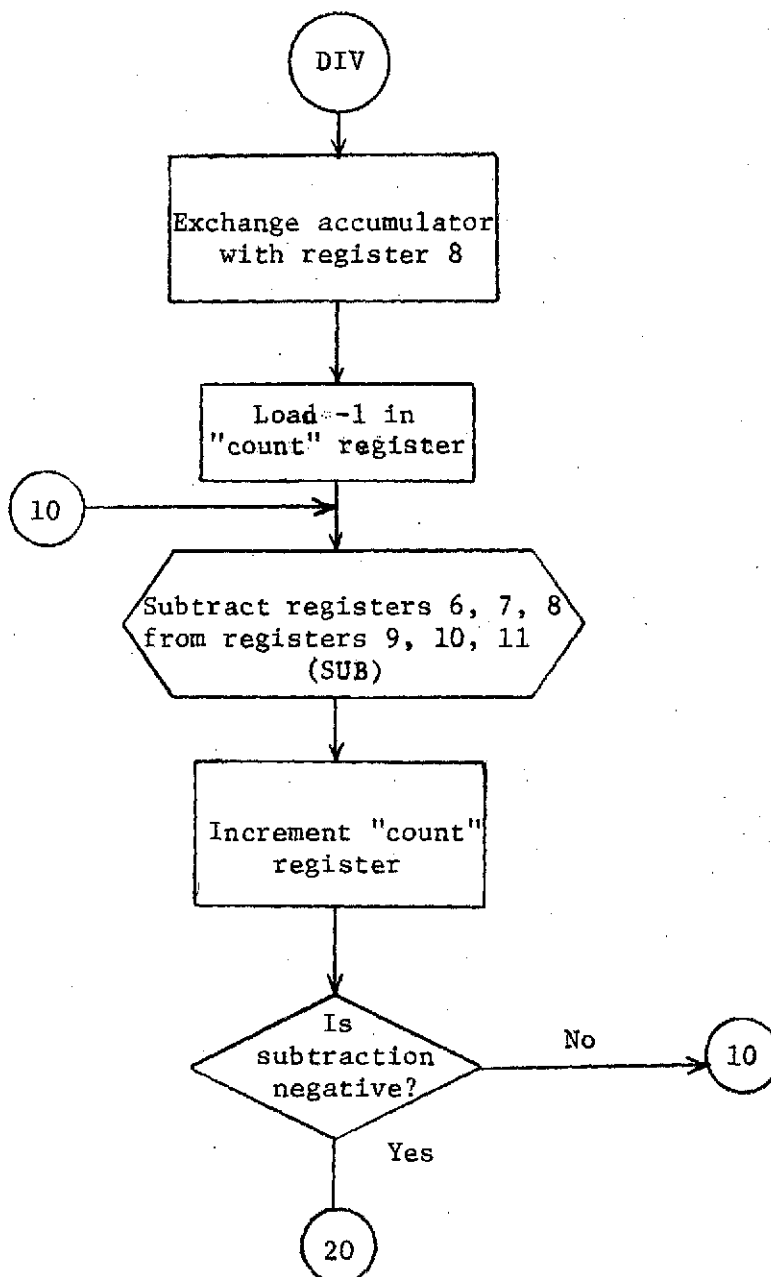
MULTIPLY SUBROUTINE CONTINUED



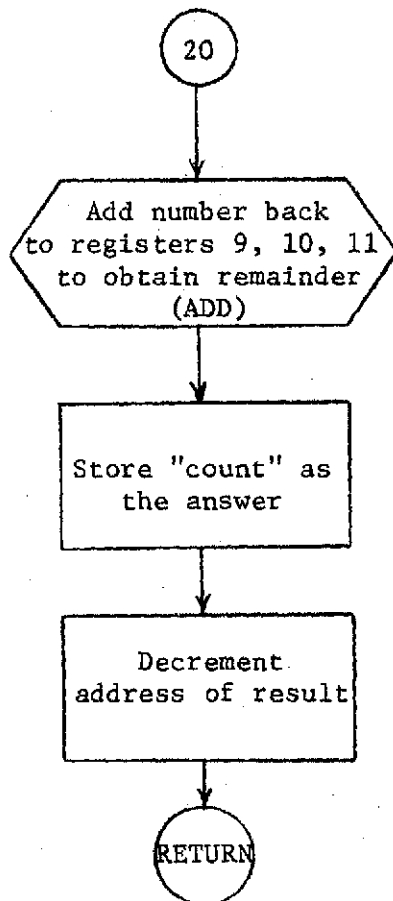
### DIVIDE SUBROUTINE

ON ENTRY: Registers 6, 7 and accumulator have values to divide into values in registers 9, 10, 11.  
Register pair 6 has the address for the result

ON EXIT: Registers 9, 10, 11 have the remainder  
Register pair 6 has been decremented



DIVIDE SUBROUTINE CONTINUED





APPENDIX B

# MAIN ROUTINE

Start	FIM	P6 φE	Load "rotary SW enable" code
	JMS	BENAB	Enable rotary SW, etc.
	FIM	P5 2φ	P5 = output bus CMD
	FIM	P4 φφ	P4 = input bus CMD
	SRC	P4	Output input CMD
	RDR		Input data on bus
	RAR		
	RAR		Shift data SW to carry
	XCH	2	Save reg 2
	JCN	CY = 1 LB5φ	Jump if data to be input
	CLB		
	WRφ		Set indicator = φ
	JMS		
		ROTPOS	Determine rotary SW pos
	XCH	2	Restore SW status
	RAR		
	FIM	P7 6φ	P7 = Starting addr of input data
	JCN	CY = 1 LB2φ	Jump if outputting input data
	FIM	P7 78	P7 = Starting addr of output data
LB2φ	JMS		Determine addr of data to be displayed
		ROTSW	
	FIM	P6 23	Addr of MS display digit
	FIM	P1 Dφ	Value = -3 for counter
LB3φ	SRC	P7	Load value to be displayed
	RDM		to regs 9,A,B from RAM
	XCH	B	
	XCH	A	
	XCH	9	
	JMS		
		INCL	
	ISZ	2 LB3φ	
	FIM	P3 8E	Reg 6,7,8 = 1000
	LDM	3	
	JMS		Determine MS digit
		DIV	
	FIM	P3 46	Reg 6,7,8 = 100

	LDM	φ	
	JMS		Determine 2nd digit
		DIV	
	FIM	P3	Reg 6,7,8 = 10
		Aφ	
	LDM	φ	
	JMS		Determine 3rd digit
		DIV	
	SRC	P6	Store L.S. digit in RAM
	LD	9	
	WRM		
	JUN		
		LB3φφ	
LB5φ	JMS		Disable "rotary SW enable"
		AENAB	
	SRC	P4	Read "indicator"
	RDφ		
	JCN	ACC ≠ 0	Jump if "indicator" = 1
		LB75	
	LDM	1	Set "indicator" = 1
	WRφ		
	FIM	P6	Reset "Char Avail" flip flop
		1C	
	JMS		
		AENAB	
	JMS		Blank BCD digit LOCS
		BLANK	
	JMS		Display blanks
		OUTSB	
LB75	FIM	P6	
		1D	
	JMS		Enable "data ready" line
		BENAB	
	SRC	P4	
	RDR		Read "data ready"
	XCH	2	Save in Reg. 2
	JMS		Disable "data ready" line
		AENAB	
	XCH	2	Restore "data ready" bit
	RAR		
	JCN	CY = 0	Jump if not ready
		LB310	
	INC	D	Increment CMD line ADR
	JMS		Enable data lines
		BENAB	
	SRC	P4	
	RDR		Input data char
	XCH	2	Save char in reg 2
	FIM	P6	
		1C	
	JMS		Disable "data" lines and reset
		AENAB	"Char Avail" FF
	CLC		
	LDM	A	ACC = "CE"

	SUB	2	
	JCN	CY = $\phi$	Jump if char was a "E"
		LB1 $\phi\phi$	
	JCN	ACC = $\phi$	Jump if char was an "CE"
		LB18 $\phi$	
	FIM	P7	P7 = Addr of BCD digits in RAM
		2 $\phi$	
LB85	SRC	P7	Shift chars left 1 and add
	RDM		new char
	XCH	2	
	WRM		
	INC	F	
	ISZ	D	
		LB85	
	JUN		
		LB3 $\phi\phi$	
LB1 $\phi\phi$	FIM	P6	
		$\phi$ E	
	JMS		Enable "Rotary switch" line
		BENAB	
	FIM	P5	
		2 $\phi$	
	JMS		Determine position of rotary switch
		ROTPOS	
	FIM	P7	
		6 $\phi$	Addr of input data
	JMS		Determine addr of data value
		ROTSW	
	FIM	P6	Determine binary from BCD
		2 $\phi$	
	FIM	P3	Regs 6,7,8 contain multiplier
		A $\phi$	Regs 9,A,B hold accumulation
	FIM	P4	
		$\phi\phi$	
	FIM	P5	
		$\phi\phi$	
	SRC	P6	Read L.S. digit
	RDM		
	XCH	9	Add to accum regs
	INC	D	
	JMS		Regs 6,7,8 = 1 $\phi$
		MUL	
	FIM	P3	
		46	
	LDM	$\phi$	Reg 6,7,8 = 1 $\phi\phi$
	JMS		
		MUL	
	FIM	P3	
		8E	
	LDM	3	Regs 6,7,8 = 1 $\phi\phi\phi$
	JMS		

		MUL	
	FIM	P1	Reg 2 = -3
		Dφ	
LB11φ	XCH	B	Store binary in RAM
	XCH	A	
	XCH	9	
	SRC	P7	
	WRM		
	JMS		Increment reg E and F if carry
		INCL	
	ISZ	2	
		LB11φ	
LB18φ	FIM	P5	P5 = Addr of BCD char
		2φ	
	JMS		Blank 4 locaitons
		BLANK	
LB3φφ	JMS		Display BCD values
		OUTSB	
LB31φ	BBL	φ	Exit

# DISPLAY SUBROUTINE

OUTSB	FIM	P1	P1 = OUTPUT command and -4 (counter)
		2C	
	FIM	P6	Output Code for L.S. Digit display
		φA	
	FIM	P2	Address of L.S. Digit (BCD)
		2φ	
OUT1φ	SRC	P2	
	RDM		Read BCD Digit
	SRC	P1	
	WRR		Output digit
	JMS		Latch digit to display
		AENAB	
	INC	D	Increment command code
	INC	5	Increment Address
	ISZ	3	
		OUT1φ	
	BBL	φ	Return

ROTARY SWITCH POSITION SUBROUTINE

ROTPOS	SRC	P5	
	LD	9	
	WRR		
	SRC	P4	
	RDR		Input status of line
	RAR		
	JCN	CY = $\phi$	Jump if switch position found
		ROTP2 $\phi$	
	INC	9	Increment index
	JUN		
		ROTPOS	
ROTP2 $\phi$	JMS		Disable "Rotary switch" line
		AENAB	
	BBL	$\phi$	Return

ROTARY SWITCH ADDRESS SUBROUTINE

ROTSW	INC	9	Register 9 has index no. (0-7)
ROT1 $\phi$	XCH	9	
	DAC		
	JCN	ACC = $\phi$	Jump if index = $\phi$
		ROT2 $\phi$	
	XCH	9	
	LDM	3	Bump address in Register pair 7
	JMS		by 3
	INC		
	JUN		
		ROT1 $\phi$	
ROT2 $\phi$	BBL	$\phi$	Return



ADDITION SUBROUTINE

ADD	CLB		
ADD1	LD	9	Add L.S. bytes
	ADD	6	
	XCH	9	Save L.S. byte
	LD	A	Add next bytes
	ADD	7	
	XCH	A	
	LD	B	Add M.S. bytes
	ADD	8	
	XCH	B	
	BBL	φ	Return

SUBTRACTION SUBROUTINE

SUB	CLB		Clear carry and accumulator
SUB1	LD	9	Subtract L.S. bytes
	SUB	6	
	CMC		Complement Carry
	XCH	9	
	LD	A	Subtract next bytes
	SUB	7	
	CMC		Complement Carry
	XCH	A	
	LD	B	Subtract M.S. bytes
	SUB	8	
	XCH	B	
	BBL	φ	Return

INCREMENT SUBROUTINE

INCL	LDM	1	Accumulator = 1
INC	CLC		Clear Carry
	ADD	F	Add Acc to Register pair 7
	XCH	F	
	LDM	$\phi$	
	ADD	E	
	XCH	E	
	BBL	$\phi$	Return

ZERO RAM SUBROUTINE

BLANK	FIM	P1	COUNTER = -4
		φC	
	LDM	φ	Clear Accumulator
BL1φ	SRC	P5	
	WRM		Write in RAM
	INC	B	Increment address
	ISZ	3	Increment counter and jump if not zero
		BL1φ	
	BBL	φ	Return

MULTIPLY SUBROUTINE

MUL	XCH	8	Move accumulator to register 8
	SRC	P6	
	RDM		Find multiplier by address in P6
	JCN	ACC = $\phi$	Jump if count = $\phi$
		MUL5 $\phi$	
MUL1 $\phi$	XCH	2	Save loop count
	JMS		Add number to accumulation area
		ADD	
	LD	2	Decrement loop count
	DAC		
	JCN	ACC $\neq$ $\phi$	Jump if not $\phi$
		MUL1 $\phi$	
MUL5 $\phi$	INC	D	Increment BCD address
	BEL	$\phi$	Return

# DIVIDE SUBROUTINE

DIV	XCH 8	Move accumulator to register 8
	LDM F	
	XCH 2	Register 2 = -1 (count)
DIV1φ	JMS	Subtract registers 6,7,8 from 9,A,B
	SUB	
	INC 2	Increment "count"
	JCN CY = 1	Jump if result not negative
	DIV1φ	
	JMS	Add registers 6,7,8 to result-remainder
	ADD1	
	SRC P6	Store "count" in BCD location
	LD 2	
	WRM	
	LD D	Decrement address
	DAC	
	XCH D	
	BBL φ	Return

## DISTRIBUTION LIST

### Copy No.

- 1 - 5    Mr. John Wells, Jr.  
         Mail Stop 470  
         National Aeronautics and Space  
         Administration  
         Langley Research Center  
         Hampton, Virginia 23365
- 6        Mr. Dennis Joe Martin  
         Office of University Affairs  
         Mail Stop 105A  
         National Aeronautics and Space  
         Administration  
         Langley Research Center  
         Hampton, Virginia 23365
- 7 - 8    E. A. Parrish, Jr.
- 9 - 10   J. H. Aylor
- 11       J. N. Warfield
- 12       I. A. Fischer
- 13 - 14   E. H. Pancake
- 15       RLES Files

PROCESSOR ARCHITECTURES UTILIZING  
MAGNETIC BUBBLE AND SEMI-CONDUCTOR MEMORIES

Interim Report

NASA Contract Number NSG-1067

Submitted to:

Mr. John Wells, Jr.  
Mail Stop 470  
National Aeronautics and Space Administration  
Langley Research Center  
Hampton, Virginia 23365

Submitted by:

E. A. Parrish, Jr.  
Associate Professor

and

J. H. Aylor  
Research Engineer

Department of Electrical Engineering  
RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES  
SCHOOL OF ENGINEERING AND APPLIED SCIENCE  
UNIVERSITY OF VIRGINIA  
CHARLOTTESVILLE, VIRGINIA

Report No. EE-4027-102-75

April 1975

Copy No. 4



## **UNIVERSITY OF VIRGINIA**

### **School of Engineering and Applied Science**

The University of Virginia's School of Engineering and Applied Science has an undergraduate enrollment of approximately 1,000 students with a graduate enrollment of 350. There are approximately 120 faculty members, of whom, about 90% hold a doctorate. Excellence in graduate education is aided and supplemented by a research program approximating \$3 million per year. This relatively high level of participation in sponsored research is one factor which helps our faculty consistently to maintain high quality graduate education at all degree levels.

As research is an integral part of the educational program, research interests parallel academic specialities. These interests range from the traditional engineering departments of Chemical, Civil, Electrical and Mechanical to include departments of Biomedical Engineering, Engineering Science & Systems, Materials Science, Nuclear Engineering, and Applied Mathematics & Computer Science. In addition to these departmental interests, there are interdepartmental groups in the areas of Automatic Controls and Applied Mechanics. All departments are authorized to offer the doctorate while the Biomedical and Materials Science Departments are graduate degree granting departments only.

The School of Engineering and Applied Science, is an integral part of an outstanding University, which has strong professional Schools of Law, Medicine, and Business Administration. In addition, the College of Arts and Sciences has strong basic science departments in Mathematics, Physics, Chemistry, and other departments relevant to the engineering research program. This not only provides an excellent scholarly climate, but also enhances the school's potential for creating truly interdisciplinary teams in the pursuit of our basic goals of education, research, and public service.

Inside this cover are listed some of the present research activities of the department from which this report originates. For more information on this or other areas, address the department chairman or Dean J. E. Gibson, Commonwealth Professor and Dean, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia 22901.